Structure-Based Self-Triggered Consensus in Networks of Multiagents with Switching Topologies

Bo Liu *Member, IEEE*, Wenlian Lu *Member, IEEE*, Licheng Jiao *Senior Member, IEEE*, and Tianping Chen *Senior Member, IEEE*

Abstract

In this paper, we propose a new self-triggered consensus algorithm in networks of multi-agents. Different from existing works, which are based on the observation of states, here, each agent determines its next update time based on its coupling structure. Both centralized and distributed approaches of the algorithms have been discussed. By transforming the algorithm to a proper discrete-time systems without self delays, we established a new analysis framework to prove the convergence of the algorithm. Then we extended the algorithm to networks with switching topologies, especially stochastically switching topologies. Compared to existing works, our algorithm is easier to understand and implement. It explicitly provides positive lower and upper bounds for the update time interval of each agent based on its coupling

structure, which can also be independently adjusted by each agent according to its own situation. Our work reveals that the event/self triggered algorithms are essentially discrete and more suitable to a discrete analysis framework. Numerical simulations are also provided to illustrate the theoretical results.

Key words: self-triggered, distributed, consensus, switching topology.

I. INTRODUCTION

Distributed control of networked cooperative multiagent systems has received much attention in recent years due to the rapid advances in computing and communication technologies. Examples include agreement or consensus problems [1]–[3], in which a group of agents seek to agree upon certain quantities of interest, formation control of robots and vehicles [4], [5], and distributed estimation [6], [7], etc.

In these applications, an important aspect is to determine the controller actuation schemes. Although continuous feedback of states have always been used in early implementations of distributed control, it is not suitable for agents equipped with embedded microprocessors that have very limited resources to transmit and gather information. To overcome this difficulty, event-triggered control scheme [8]–[16] was proposed to reduce the controller updates. In fact, event-driven strategies for multi-agent systems can be viewed as a linearization and discretization process, which was considered and investigated in early papers [17], [18]. For example, in the paper [17], following algorithm was investigated

$$x^i(t+1) = f(x^i(t)) + c_i \sum_{j=1}^{m} a_{ij}(f(x^j(t))) \tag{1}$$

which can be considered as nonlinear consensus algorithm. In particular, let $f(x(t)) = x(t)$ and $c_i = (t_{k+1}^i - t_k^i)$. Then (1) becomes

$$x^i(t_{k+1}^i) = x^i(t_k^i) + (t_{k+1}^i - t_k^i) \sum_{j=1}^{m} a_{ij} x^j(t_k^i), \tag{2}$$

which is some variant of the event triggering (distributed, self triggered) model for consensus problem. In centralized control, the bound for $(t_{k+1}^i - t_k^i) = (t_{k+1} - t_k)$ to reach synchronization was given in the paper [17] when the coupling graph is indirected and in [18] for the directed coupling graph. The key point in event-triggered algorithm is the design of a decision maker that

2

determines when the next actuator update should occur. The existing event-triggered algorithms are all based on the observation of states. For example, in a typical event-triggered algorithm proposed in [8], an update is triggered when a certain error of the states becomes large enough with respect to the norm of the state, which requires a continuous observation of the states. In addition, self-triggered control [19]–[23] has been proposed as a natural extension of event-triggered control, in which each agent predicts its next update time based on discontinuous state observation to further reduce resource usage for the control systems.

Both the event-triggered and self-triggered control algorithms that have been proposed till now have some drawbacks in common. First, the resulting system in event-triggered control is one that with system delays, especially with self delays, which generally is difficult to handle. And the existing analysis for such algorithms are always based on some quadratic Lyapunov function, which has very strict restrictions on the network structure. For example, to the best of our knowledge, the latest analysis is still restricted to static networks. Second, since these algorithms are based on the observation of states, which are generally much complicated and untraceable, making it difficult to predict and exclude some unexpected possibilities such as the occurrence of zero inter-execution times.

To overcome such difficulties, we proposed a new kind of self-triggered consensus algorithms that are structure-based. That is, each agent predicts its next update time based on its coupling structure with its neighbours instead of the observation of their states. This can bring several advantages. First, since the coupling structure is relatively simpler and more traceable than the states, it is relatively easier to handle. Actually, in our algorithm, the occurrence of zero inter-execution time has been excluded by directly providing a positive lower bound on the update intervals.

Second, although the resulting system of the self-triggered algorithm is one with self delays, by some proper transformation, we showed that the system can corresponding to a discrete-time system without self-delays. We have proposed the algorithm in both centralized and distributed approaches. In the centralized approach, the system can be directly related to a discrete-time system without delays, which can be seen as a discrete version of the nominal system. However, in the distributed approach, the situation is much more complicated and the system can not be directly transformed into its discretized version. However, by some proper indirect transformation, we showed that the convergence of the nominal system can be reduced to that of a

3

discrete-time consensus algorithm with off-diagonal delays but without self delays. Thus, other than using a quadratic Lyapunov function, the convergence of the original algorithm can be solved by the analysis of a discrete-time consensus algorithm. This brings another possibility that is also considered as a big advantage and a major contribution of this work. That is, we can analyze networks with switching topologies, especially stochastically switching topologies.

The rest of the paper is organized as follows: Section II provides some preliminaries on matrix and graph theory that will be used in the main text. Section III provides the self-triggered consensus algorithm in both centralized and distributed approach with convergence analysis. Section IV provide an example with numerical simulation to illustrate the theoretical results obtained in the previous section. The paper is concluded in Section V.

II. PRELIMINARIES

In this section, we present some definitions and results in matrix and graph theories and probability theories that will be used later.

For a matrix $A = [a_{ij}]$, a_{ij} represents the entry of A on the ith row and jth column, which is sometimes also denoted as $[A]_{ij}$. A matrix $A = [a_{ij}]$ is called a *nonnegative matrix* if $a_{ij} \geq 0$ for all i, j. And A is called a *stochastic matrix* if A is square and $\sum_j a_{ij} = 1$ for each i. Given a nonnegative matrix A and $\delta > 0$, the *δ-matrix* of A is a matrix that has nonzero entries only in the place that A has entries equal to or greater than δ. For two matrix A, B of the same dimension, we write $A \geq B$ if $A - B$ is a nonnegative matrix. Throughout this paper, we use $\prod_{i=1}^{k} A_i = A_k A_{k-1} \cdots A_1$ to denote the left product of matrices.

A *directed graph* $\mathcal{G}$ is defined by its vertex set $\mathcal{V}(\mathcal{G}) = \{v_1, \cdots, v_n\}$ and edge set $\mathcal{E}(\mathcal{G}) \subseteq \{(v_i, v_j) : v_i, v_j \in \mathcal{V}(\mathcal{G})\}$, where an edge is an ordered pair of vertices in $\mathcal{V}(\mathcal{G})$. If $(v_i, v_j) \in \mathcal{E}(\mathcal{G})$, then v_i is called the *(in-)neighbour* of v_j. $\mathcal{N}_i = \{j : (v_j, v_i) \in \mathcal{E}(\mathcal{G})\}$ is the set of neighbours of agent i. A *directed path* in $\mathcal{G}$ is an ordered sequence of vertices $v_1, \cdots, v_s$ such that $(v_i, v_{i+1}) \in \mathcal{E}(\mathcal{G})$ for $i = 1, \cdots, s-1$. A *directed tree* is a directed graph where every vertex has exactly one in-neighbour except one ,called the *root*, without any in-neighbour. And there exists a directed path from the root to any other vertex of the graph. A subgraph of $\mathcal{G}$ is a directed graph $\mathcal{G}_S$ satisfying $\mathcal{V}(\mathcal{G}_S) \subseteq \mathcal{V}(\mathcal{G})$, and $\mathcal{E}(\mathcal{G}) \subseteq \mathcal{E}(\mathcal{G})$. A *spanning subgraph* of $\mathcal{G}$ is a subgraph of $\mathcal{G}$ that has the same vertex set with $\mathcal{G}$. We say $\mathcal{G}$ has a *spanning tree* if $\mathcal{G}$ has a spanning subgraph that is a tree.

4

A *weighted directed graph* is a directed graph where each edge is equipped with a weight. Thus, a graph $\mathcal{G}$ of n vertices corresponds to an $n \times n$ nonnegative matrix $A = [a_{ij}]$, called the *weight matrix* in such way that $(v_j, v_i) \in \mathcal{E}(\mathcal{G})$ if and only if $a_{ij} > 0$. On the other hand, given an $n \times n$ nonnegative matrix A, it corresponds to a weighted directed graph $\mathcal{G}(A)$ such that $\mathcal{G}(A)$ has A as its weight matrix. Using this correspondence, we can introduce the concept of δ-graph. The δ-*graph* of $\mathcal{G}$ is a weighted directed graph that has the δ-matrix of $\mathcal{G}$'s weight matrix as its weight matrix. A graph $\mathcal{G}$ has a δ *spanning tree* if its δ-graph has a spanning tree.

From the weight matrix $A = [a_{ij}]$ of a graph $\mathcal{G}$, we define its graph Laplacian $L = [l_{ij}]$ as follows:

$$
l_{ij} = \begin{cases} \sum_{k \neq i} a_{ik}, & i = j; \\ -a_{ij}, & i \neq j. \end{cases}
$$

Thus L has zero row sum with $l_{ii} > 0$ while $l_{ij} \leq 0$ for $i, j = 1, 2, \cdots, n$ and $i \neq j$. And the nonnegative linear combination of several graph Laplacians is also a graph Laplacian of some graph. There is a one-to-one correspondence between a graph and its weight matrix or its Laplacian matrix. In the following, for the sake of simplicity in presentation, sometimes we don't explicitly distinguish a graph from its weight matrix or Laplacian matrix, i.e., when we say a matrix has some property that is usually associated with a graph, what we mean is that property is held by the graph corresponding to this matrix. For example, when we say a nonnegative matrix A has a spanning tree, what we mean is that the graph of A has a spanning tree. And it is of similar meaning when we say that a graph has some property that is usually associated with a matrix.

Let $\{\Omega, \mathcal{F}, \mathbb{P}\}$ be a probability space, where Ω is the sample space, $\mathcal{F}$ is σ-algebra on Ω, and $\mathbb{P}$ is the probability on $\mathcal{F}$. We use $\mathbb{E}\{\cdot\}$ to denote the mathematical expectation and $\mathbb{E}\{\cdot|\mathcal{F}\}$ the conditional expectation with respect to $\mathcal{F}$, i.e., $\mathbb{E}\{\cdot|\mathcal{F}\}$ is a random variable that is measurable with respect to $\mathcal{F}$.

Definition 1 (adapted process/sequence): Let $\{A_k\}$ be a stochastic process defined on the basic probability space $\{\Omega, \mathcal{F}, \mathbb{P}\}$, and let $\{\mathcal{F}_k\}$ be a filtration, i.e., a sequence of nondecreasing sub-σ-algebras of $\mathcal{F}$. If A_k is measurable with respect to $\mathcal{F}_k$, then the sequence $\{A_k, \mathcal{F}_k\}$ is called an adapted process/sequence.

Remark 1: For example, let $X(\omega)$ be a random variable defined on Ω, then $\{\mathbb{E}\{X(\omega)|\mathcal{F}_k\}, \mathcal{F}_k\}$ is an adapted sequence.

III. SELF-TRIGGERED CONSENSUS ALGORITHM

In a network of n agents, with $x_i \in \mathbb{R}$ being the state of agent i, we assume that each agent's dynamics obey a single integrator model

$$\dot{x}_i(t) = u_i(t), \quad i = 1, 2, \cdots, n, \tag{3}$$

where $u_i(t)$ is the control law for agent i at time t. In consensus algorithm, the control law is usually given by([1], [29], [30])

$$u_i(t) = -\sum_{j \in \mathcal{N}_i} l_{ij}[x_j(t) - x_i(t)] = -\sum_{j=1}^{n} l_{ij}[x_j(t) - x_i(t)]. \tag{4}$$

Thus the consensus algorithm can be written as

$$\dot{x}_i(t) = -\sum_{j=1}^{n} l_{ij}[x_j(t) - x_i(t)], \tag{5}$$

or in matrix form as

$$\dot{x}(t) = -Lx(t), \tag{6}$$

where $x(t) = [x_1(t), \cdots, x_n(t)]^\top \in \mathbb{R}^n$ is the stack vector of the agents' states. In event/self-triggered control, the control law $u(t)$ is piecewise constant, i.e.,

$$u(t) = u(t_k), \ t \in [t_k, t_{k+1}), \tag{7}$$

where $\{t_k\}$ is the time sequence that an update of $u(\cdot)$ occurs. The central point of such algorithm is the choice of appropriate $\{t_k\}$ such that some desired properties of the algorithm such as stability and convergence can be preserved. This can be done in a centralized approach or a distributed approach, both of which will be discussed in the following.

A. Centralized Approach

In this subsection, we first consider centralized self-triggered consensus algorithm. The sequence of the update time is denoted by: $t_0, t_1, t_2, \cdots$, then the self-triggered algorithm has the form:

$$\dot{x}(t) = -Lx(t_k), \ t \in [t_k, t_{k+1}). \tag{8}$$

Denote $\Delta t_k = t_{k+1} - t_k$ for $k = 0, 1, \cdots$. We have the following Theorem.

Theorem 1: If the graph $\mathcal{G}(L)$ has a spanning tree and there exists $\delta \in (0, 1/2)$ such that $\delta/l_{\max} \leq \Delta t_k \leq (1-\delta)/l_{\max}$ for each k, where $l_{\max} = \max_i\{l_{ii}\}$, then the algorithm will achieve consensus asymptotically.

Proof: From the algorithm, for each $t \in (t_k, t_{k+1}]$, since

$$\dot{x}(t) = -Lx(t_k),$$

we have

$$x(t) = x(t_k) - (t - t_k)Lx(t_k) = [I - (t - t_k)L]x(t_k).$$

Particularly, let $t = t_{k+1}$, we have

$$x(t_{k+1}) = (I - \Delta t_k L)x(t_k) = A_k x(t_k),$$

where $A_k = I - \Delta t_k L$. It is easy to verify that A_k is a stochastic matrix for each k. In fact, since $\delta/l_{\max} \leq \Delta t_k \leq (1-\delta)/l_{\max}$,

$$[A_k]_{ii} = 1 - \Delta t_k l_{ii} \geq 1 - \Delta t_k l_{\max} \geq 1 - \frac{1-\delta}{l_{\max}}l_{\max} = \delta.$$

For $i \neq j$,

$$[A_k]_{ij} = -\Delta t_k l_{ij} \geq 0.$$

It implies that $[A_k]_{ij}$ is positive if and only if $-l_{ij}$ is positive, and the positive elements of $[A_k]$ is uniformly lower bounded by a positive scalar since both Δt_k and $-l_{ij}$ is uniformly lower bounded by a positive scalar. Furthermore, $\sum_{j=1}^{n}[A_k]_{ij} = 1 - \Delta t_k \sum_{j=1}^{n} l_{ij} = 1$. By a standard argument from the theory of products of stochastic matrices, there exists $x^* \in \mathbb{R}$ such that

$$\lim_{k \to +\infty} x_i(t_k) = x^*.$$

On the other hand, it is not difficult to verify that the function $\max_i\{x_i(t)\}$ is nonincreasing and $\min_i\{x_i(t)\}$ is nondecreasing. Thus,

$$\lim_{t \to +\infty} x_i(t) = x^*.$$

■

Using a similar argument as that in Theorem 1, it is not difficult to propose a centralized event-triggered algorithm on networks with stochastic switching topologies. For example, consider the following consensus algorithm:

$$\dot{x}(t) = -L^k x(t_k), \ t \in [t_k, t_{k+1}), \tag{9}$$

where $L^k = [l_{ij}^k]$ is uniformly bounded in k and $l_{\max}^k = \max_i\{l_{ii}^k\} \in [l_{\min}, l_{\max}]$ for some $0 < l_{\min} < l_{\max}$. Before provide the next theorem, we first summarize the following assumption.

Assumption 1: (i) There exists $\delta \in (0, 1/2)$ such that $\Delta t_k \in [\delta/l_{\max}^k, (1 - \delta)/l_{\max}^k]$;

(ii) $\{L^k, \mathcal{F}_k\}$ is an adapted random sequence, and there exists $h > 0$, $\delta' > 0$ such that the graph corresponding to the conditional expectation $\mathbb{E}\{\sum_{m=k+1}^{k+h} L^m | \mathcal{F}_k\}$ has a δ'-spanning tree;

Now we have

Theorem 2: Under assumption1, the self-triggered consensus algorithm (9) will reach consensus almost surely.

Proof: Similar as in the proof of Theorem 1, we have $x(t_{k+1}) = A_k x(t_k)$ with $A_k = I - \Delta t_k L^k$. Since $\{\Delta t_k\}$ is a deterministic sequence, $\{A_k, \mathcal{F}_k\}$ is also an adapted sequence, and

$$
\begin{aligned}
\mathbb{E}\{\sum_{m=k+1}^{k+h} A_m | \mathcal{F}_k\} &= hI - \mathbb{E}\{\sum_{m=k+1}^{k+h} \Delta t_m L^m | \mathcal{F}_k\} \\
&\geq h\delta I - \frac{\delta}{l_{\max}} \mathbb{E}\{\sum_{m=k+1}^{k+h} L_0^m | \mathcal{F}_k\},
\end{aligned}
$$

where $[L_0^m]_{ij} = [L^m]_{ij}$ for $i \neq j$ and $[L_0^m]_{ii} = 0$ for all i. Thus, $\mathbb{E}\{\sum_{m=k+1}^{k+h} A_m | \mathcal{F}_k\}$ has a δ''-spanning tree with $\delta'' = \delta'\delta/l_{\max} > 0$. From Theorem 3.1 of [28], the sequence $\{x(t_k)\}$ will reach consensus almost surely, thus $x(t)$ will also reach consensus almost surely. $\blacksquare$

As corollaries in special cases, it can be shown that almost sure consensus still holds if we replace item (ii) in Assumption 1 with one of the following more special ones.

(ii') $\{L^k\}$ is an independent and identically distributed sequence, and there exists $\delta' > 0$ such that the graph corresponding to the expectation $\mathbb{E}\,L^k$ has a δ'-spanning tree;

(ii'') $\{L^k\}$ is a homogenous Markov chain with a stationary distribution π, and there exists $\delta' > 0$ such that the graph corresponding to the expectation with respect to π, $\mathbb{E}_\pi L^k$, has a δ'-spanning tree;

B. Distributed Approach

In this section, we discuss the distributed event-triggered consensus algorithm.

Let $\{t_i^k\}_{k=0}^{+\infty}$ be the time sequence such that t_i^k is the kth time that agent i updates its control

law. Then the distributed self-triggered algorithm has the following form:

$$x_i(t) = -\sum_{j=1}^{n} l_{ij} x_j(t_j^{k_j(t)}), \quad i = 1, \cdots, n, \tag{10}$$

where $k_j(t) = \max\{k : t_j^k \le t\}$.

Denote $\Delta t_i^k = t_i^{k+1} - t_i^k$. Then we have the following Theorem.

Theorem 3: If the graph $\mathcal{G}(L)$ has a spanning tree and there exists $\delta_i \in (0, 1/2)$ for $i = 1, 2, \cdots, n$ such that $\Delta t_i^k \in [\delta_i/l_{ii}, (1-\delta_i)/l_{ii}]$, then the distributed event triggered algorithm can reach a consensus as $t \to \infty$.

The proof will be divided into two steps corresponding to two lemmas. Now we give and prove the first lemma.

Lemma 1: If there exists $x^* \in \mathbb{R}$ such that

$$\lim_{k \to \infty} x(t_i^k) = x^*, \quad i = 1, 2, \cdots, n,$$

then the distributed event-triggered algorithm will reach a consensus, i.e.,

$$\lim_{t \to \infty} x_i(t) = x^*, \quad i = 1, 2, \cdots, n.$$

Proof: By the definition of the algorithm, for agent i,

$$\dot{x}_i(t) = -\sum_{j=1}^{n} l_{ij} x(t_j^{k_j(t)}) = -\sum_{j=1, j \ne i}^{n} l_{ij}[x_j(t_j^{k_j(t)}) - x_i(t_i^{k_i(t)})].$$

Since $\lim_{t \to \infty} k_j(t) = +\infty$,

$$\lim_{t \to \infty} \dot{x}_i(t) = -\sum_{j=1, j \ne i}^{n} l_{ij}(x^* - x^*) = 0.$$

Thus,

$$\begin{aligned}
\lim_{t \to \infty} |x_i(t) - x^*| &\le \lim_{t \to \infty} |x_i(t) - x_i(t_i^{k_i(t)})| + \lim_{t \to \infty} |x_i(t_i^{k_i(t)}) - x^*| \\
&\le \lim_{t \to \infty} \Delta t_i^{k_i(t)} \max_{s \in [t_i^{k_i(t)}, t]} |\dot{x}_i(s)| + \lim_{t \to \infty} |x_i(t_i^{k_i(t)}) - x^*| \\
&\le \frac{1 - \delta_i}{l_{ii}} \lim_{t \to \infty} \max_{s \in [t_i^{k_i(t)}, t]} |\dot{x}_i(s)| + \lim_{t \to \infty} |x_i(t_i^{k_i(t)}) - x^*| \\
&= 0.
\end{aligned}$$

∎

Now we are to provide and prove the second lemma.

Lemma 2: Under the assumption in Theorem 3, there exists x^* such that

$$\lim_{k \to \infty} x_i(t_i^k) = x^*.$$

Proof: Let $\{t_k\}$ be the time sequence that the kth update occurred in the network, i.e., $\{t_k\} = \cup_{i=1}^n \{t_i^{k'}\}$, and $\Delta t_k = t_{k+1} - t_k$. In case for that $t_k = t_i^{k'} = t_j^{k''}$ for some some k, k', k'' and $i \neq j$, we consider the update of agent i and j at time t_k as one update of the network. Now, construct an auxiliary sequence $\{y(k)\}$, where $y(k) = [y_1(k), y_2(k), \cdots, y_n(k)]^\top \in \mathbb{R}^n$ with $y_i(k) = x_i(t_i^{k_i(t_k)})$, i.e., $y_i(k)$ is the latest state that agent i uses in its control law at time t_k. Now consider the evolution of $\{y(k)\}$.

Case 1. Agent i does not update at time t_{k+1}: In this case, by definition,

$$y_i(k+1) = y_i(k);$$

Case 2. Agent i does update at time t_{k+1}: In this case, $t_i^{k_i(t_{k+1})} = t_{k+1}$, assume $t_i^{k_i(t_k)} = t_{k-d_{ik}}$ be the last update of agent i before t_{k+1}, with $d_{ik} \geq 0$ being the number of updates occur at all other agents x_j, $j \neq i$, between the two successive updates of agent i, i.e., in the interval $(t_i^{k_i(t_k)}, t_i^{k_i(t_{k+1})})$. By definition $y_i(k) = \cdots = y_i(k - d_{ik})$, and $\dot{x}_i(t) = -\sum_{j=1}^n l_{ij} y_j(k')$ for $t \in (t_{k'}, t_{k'+1})$ for all k'. Since $[t_i^{k_i(t_k)}, t_i^{k_i(t_{k+1})}) = \cup_{m=k-d_{ik}}^k [t_m, t_{m+1})$,

$$
\begin{aligned}
y_i(k+1) &= x_i(t_{k+1}) = x_i(t_i^{k_i(t_k)}) + \int_{t_i^{k_i(t_k)}}^{t_i^{k_i(t_{k+1})}} \dot{x}_i(t)dt \\
&= x_i(t_{k-d_{ik}}) + \sum_{m=0}^{d_{ik}} \int_{t_{k-d_{ik}+m}}^{k-d_{ik}+m+1} \dot{x}_i(t)dt \\
&= x_i(t_{k-d_{ik}}) - \sum_{m=0}^{d_{ik}} \Delta t_{k-d_{ik}+m} \sum_{j=1}^{n} l_{ij} y_j(k-d_{ik}+m) \\
&= y_i(k-d_{ik}) - \sum_{m=0}^{d_{ik}} \Delta t_{k-d_{ik}+m} l_{ii} y_i(k-d_{ik}+m) \\
&\quad - \sum_{m=0}^{d_{ik}} \Delta t_{k-d_{ik}+m} \sum_{j=1,j\neq i}^{n} l_{ij} y_j(k-d_{ik}+m) \\
&= y_i(k) - \sum_{m=0}^{d_{ik}} \Delta t_{k-d_{ik}+m} l_{ii} y_i(k) - \sum_{m=0}^{d_{ik}} \Delta t_{k-d_{ik}+m} \sum_{j=1,j\neq i}^{n} l_{ij} y_j(k-d_{ik}+m) \\
&= (1 - l_{ii} \sum_{m=0}^{d_{ik}} \Delta t_{k-d_{ik}+m}) y_i(k) - \sum_{m=0}^{d_{ik}} \sum_{j=1,j\neq i}^{n} \Delta t_{k-d_{ik}+m} l_{ij} y_j(k-d_{ik}+m) \\
&= (1 - \Delta t_i^{k_i(t_k)} l_{ii}) y_i(k) - \sum_{m=0}^{d_{ik}} \sum_{j=1,j\neq i}^{n} \Delta t_{k-d_{ik}+m} l_{ij} y_j(k-d_{ik}+m) \\
&= \sum_{m=0}^{d_{ik}} \sum_{j=1}^{n} a_{ij}^m(k) y_j(k-m),
\end{aligned}
$$

where $a_{ii}^0(k) = 1 - \Delta t_i^{k_i(t_k)} l_{ii}$, $a_{ii}^m(k) = 0$ for $m = 1, 2, \cdots, d_{ik}$ and $a_{ij}^m(k) = -\Delta t_{k-m} l_{ij}$. It is easy to verify that $a_{ii}^0(k) = 1 - \Delta t_i^{k_i(t_k)} l_{ii} \geq 1 - (1 - \delta_i) = \delta_i$, $a_{ij}^m = -\Delta t_{k-m} l_{ij} \geq 0$, and

$$
\sum_{m=0}^{d_{ik}} \sum_{j=1}^{n} a_{ij}^m(k) = 1.
$$

Besides, by the assumption $\delta_i/l_{ii} \leq \Delta t_i^k \leq (1 - \delta_i)/l_{ii}$ for each i, k, it is clear that in the interval $[t_i^k, t_i^{k+1}]$, only finite updates occur for agents x_j with $j \neq i$, and the number of updates is uniformly upper bounded, i.e., there exists $\tau > 0$ independent of i, k, such that $d_{ik} \leq \tau$. For example, let $\delta_{\min} = \min_i\{\delta_i/l_{ii}\}$, and $\delta_{\max} = \max_i\{\delta_i/l_{ii}\}$, then $\delta_{\min} \leq \Delta t_i^k \leq \delta_{\max}$ for all i, k, and on each time interval of length $m\delta_{\min}$, at most $m + 1$ updates occur for each agent. Let h' be the smallest positive integer satisfying

$h'\delta_{\min} \geq \delta_{\max}$, then on each time interval of length $\delta_{\max}$, at most $h'+1$ updates occur for each agent. Thus on the interval (t_i^k, t_i^{k+1}), at most $(n-1)(h'+1)$ updates occur for all agents j with $i \neq j$. Pick $\tau = (n-1)(h'+1)$, then $d_{ik} \leq \tau$ for all i, k.

If agent i does not update at time t_{k+1}, i.e., case 1, we define $a_{ij}^m(k) = 0$ for all $m = 0, 1, \cdots, \tau$ and $i, j = 1, 2, \cdots, n$ except for $a_{ii}^0(k) = 1$,

If agent i updates at time t_{k+1}, i.e., case 2, we define $a_{ij}^m(k) = 0$ for all $i, j = 1, 2, \cdots, n$ and $m = d_{ik}+1, \cdots, \tau$ when $d_{ik} < \tau$.

For both cases, we can give a uniform iterative formula for $y(k)$ as:

$$y(k+1) = \sum_{l=0}^{\tau} \sum_{j=1}^{n} a_{ij}^l(k) y_j(k-l), \tag{11}$$

where $a_{ij}^l(k) \geq 0$ for all i, j, l, k, and

$$\sum_{l=0}^{\tau} \sum_{j=1}^{n} a_{ij}^l(k) = 1,$$

for all i, k. Denote $\delta = \min_i\{\delta_i\} > 0$, we have $a_{ii}^0(k) \geq \delta$ for all i, k.

Construct a new matrix $B(k) = [b_{ij}(k)]$, where

$$b_{ii}(k) = a_{ii}^0(k) \tag{12}$$

and

$$b_{ij}(k) = \sum_{l=0}^{\tau} a_{ij}^l(k), \quad if \ i \neq j \tag{13}$$

Obviously, $B(k) \geq \delta I$, and $b_{ij}(k) = \Delta t_i^{k_i(t_k)} l_{ij} \geq \delta_{\min} l_{ij}$ if agent i updates at time t_{k+1}. If there exists $h > 0$ such that on each interval $[t_k, t_{k+h}]$ all agents update at least once, then $\sum_{m=k}^{k+h} B(m) \geq -\delta_{\min} L$ for each k. Since $\mathcal{G}(L)$ has a spanning tree, there exists $\delta' > 0$ such that $\sum_{m=k}^{k+h} B(m)$ has a δ'-spanning tree for each k. From Corollary 2 in the appendix, the sequence $\{y(k)\}$ will reach a consensus, i.e., there exists $x^* \in \mathbb{R}$ such that

$$\lim_{k \to \infty} y_i(k) = x^*.$$

This is equivalent to

$$\lim_{k \to \infty} x_i(t_i^k) = x^*.$$

At last, we give some hint on how to calculate the quantity h mentioned above. It is easy to see that each agent updates at least once on each time interval of length $\delta_{\max}$, since $\Delta t_i^k \leq \delta_{\max}$ for

12

all i, k. And in each time interval of length $\delta_{\max}$, there are at most $n(h'+1)$ updates of all the agents, where h' is defined as before. Pick $h = n(h'+1)$, then for each k, each agent updates at least once in the time interval $[t_k, t_{k+h}]$. ∎

Remark 2: It is not difficult to see that this algorithm is applicable to leader-follower networks, in which the leader receives no information from others. For example, if agent i is the leader, then $l_{ii} = 0$ and the update time interval $\Delta t_i^0 = +\infty$ by definition, which mean the leader i does not need to update its state. Generally, if agent i wants a long update time interval, all it needs to do is to choose small coupling weights so that l_{ii} is small, and this can be done independent of all other agents.

From the above analysis, it is not difficult to see that the distributed event-triggered algorithm can also be applied to networks with switching topologies. For example, at each update time t_i^k, agent i may reset its coupling weights l_{ij} such that the algorithm can be written in the form:

$$\dot{x}_i(t) = -\sum_{j=1}^{n} l_{ij}^k x_j(t_j^{k_j(t)}), \quad t \in [t_i^k, t_i^{k+1}), \ i = 1, 2, \cdots, n, \tag{14}$$

where $\delta_i / l_{ii}^k \leq \Delta t_i^k \leq (1 - \delta_i)/l_{ii}^k$. One of the great advantage of this algorithm is that each agent can adjust its update time interval independently based on its own need. If agent i wants a longer update time interval, then it may decrease the coupling weight, otherwise, increase the coupling weight. For example, each agent i may store a scaling factor ϵ_i^k such that

$$l_{ij}^k = \epsilon_i^k l_{ij}. \tag{15}$$

By a similar analysis as in the case of fixed topology, we can prove

Theorem 4: Suppose that $\epsilon_{\min}$ and $\epsilon_{\max}$ are two positive number satisfying $0 < \epsilon_{\min} < \epsilon_{\max}$ and for all i, k, $\epsilon_{\min} \leq \epsilon_i^k \leq \epsilon_{\max}$. If the graph $\mathcal{G}(L)$ has a spanning tree, then the distributed event-triggered consensus algorithm with switching topology (14) and weight updating rule (15) can reach consensus as $t \to \infty$.

Finally, we investigate event-triggered consensus algorithm in networks with stochastically switching topologies in which each agent i updates its coupling weights independently at the same time it updates its state.

Given $0 < a < b$, and let $S_i(a, b) = \{s = [s_1, \cdots, s_n] : s_i \in [a, b], s_j \leq 0, j \neq i, \sum_{j=1}^{n} s_j = 0\}$. The event-triggered algorithm can be formulated as follows:

$$\dot{x}_i(t) = -\sum_{j=1}^{n} \tilde{l}_{ij}^k x_j(t_j^{k_j(t)}), \ t \in [t_i^k, t_i^{k+1}), \tag{16}$$

13

where $\tilde{L}_i^k = [\tilde{l}_{i1}^k, \cdots, \tilde{l}_{in}^k] \in S_i(a_i, b_i)$ for some given $0 < a_i < b_i$, and $\{\tilde{L}_i^k\}$ is an independent and identically distributed sequence. Before stating the convergence result, we summarize the following assumption.

Assumption 2: (i) There exist $\delta_i \in (0, 1/2)$ such that $\Delta t_i^k \in [\delta_i/\tilde{l}_{ii}^k, (1-\delta_i)/\tilde{l}_{ii}^k]$;

(ii) There exists $\delta > 0$ such that the graph with Laplacian being $\mathbb{E}\,\tilde{L}^k$ has a δ-spanning tree, where $\tilde{L}^k = [\tilde{L}_1^{k\top}, \cdots, \tilde{L}_n^{k\top}]^\top$ is a matrix with its ith row being $\tilde{L}_i^k$;

Theorem 5: Under Assumption 2, the event-triggered consensus algorithm (16) will reach a consensus almost surely.

Proof: The algorithm (16) can be reformulated as:

$$\dot{x}_i(t) = -\sum_{j=1}^{n} l_{ij}^k x_j(t_j^{k_j(t)}), \ t \in [t_k, t_{k+1}), \tag{17}$$

where $l_{ij}^k = \tilde{l}_{ij}^{k_i(t_k)}$. Let $L^k = [l_{ij}^k]$, then generally the sequence $\{L^k\}$ is not an independent sequence since for each k, $l_{ij}^k = l_{ij}^{k+1}$ if no update of agent i occurs at time t_{k+1}. However, similar to the analysis given above, it can be seen that L^k is independent of $L^{k'}$ for $k' \geq k + N$, where $N = n(h'+1)$ and h' is the smallest positive integer satisfying $h' \min_i\{\delta_i/b_i\} \geq \max_i\{(1-\delta_i)/a_i\}$.

Similarly, we have

$$y(k+1) = \sum_{l=0}^{N}\sum_{j=1}^{n} a_{ij}^l(k)y_j(k-l), \tag{18}$$

where $a_{ij}^l(k) \geq 0$ for all i, j, l, k, and

$$\sum_{l=0}^{N}\sum_{j=1}^{n} a_{ij}^l(k) = 1,$$

for all i, k. Since $a_{ij}^l(k) = -\Delta t_{k-l}l_{ij}^{k-l}$, we can see that $A^0(k), \cdots, A^N(k)$ are independent of $A^0(k'), \cdots, A^N(k')$, when $k' \geq k + 2N$. On the other hand, in each time interval (t_k, t_{k+N}), each agent there at least updates once. Thus we have

$$\sum_{m=k+1}^{k+N} \mathbb{E}\,B(k) \geq \delta'\tilde{L}_0,$$

where $B(k)$ is defined as before and $\delta' = \min_{i,k}\{\Delta t_i^k\} = \min_i\{\delta_i/b_i\} > 0$, and $[\tilde{L}_0]_{ij} = -[\mathbb{E}\,\tilde{L}^k]_{ij}$ for $i \neq j$ and $[\tilde{L}_0]_{ii} = 0$ for all i. This implies that $\sum_{m=k+1}^{k+N} \mathbb{E}\,B(k)$ has a δ''-spanning tree with $\delta'' = \delta\delta' > 0$. The conclusion follows from Corollary 3 in the Appendix.

14

IV. NUMERICAL SIMULATION

In this section, we provide an example to illustrate the theoretical results in the previous section. We use the distributed event-triggered algorithm in networks with stochastically switching topologies as considered in Theorem 5.

Consider a network of four agents with

$$\tilde{L}_1^k \in \{[1, -1, 0, 0], [1, 0, 0, -1]\},$$

$$\tilde{L}_2^k = \{[-1, 1, 0, 0], [0, 1, -1, 0]\},$$

$$\tilde{L}_3^k \in \{[0, -1, 1, 0], [0, 0, 1, -1]\},$$

$$\tilde{L}_4^k \in \{[0, -1, 0, 1], [0, 0, -1, 1]\},$$

and each agent selects its coupling weights using a uniform distribution. We choose $\delta_i = 0.1$ for all the agents. The next update time is randomly chosen from the permissible range. It is not difficult to verify that the conditions in Theorem 5 are satisfied. The simulation results are provided in Fig. 1 with the initial value of the four agents being randomly chosen. It can be seen that the agents actually reached consensus.

V. CONCLUSION

In this paper, we proposed a new structure-based self-triggered consensus algorithm in both centralized approach and distributed approach. Different from existing works that used a quadratic Lyapunov function as the analysis tool, we reduce the self-triggered consensus algorithm to a discrete-time consensus algorithm by some proper transformation, which enables the application of the product theory of stochastic matrices to the convergence analysis. Compared to existing work, our method provides several advantages. First, each agent does not need to calculate the system error to determine its update time. Second, it provides explicit positive lower and upper bounds for the update interval of each agent based on its coupling weights, which can be adjusted by each agent independent of other agents. We also used our method to investigate networks with switching topologies, especially networks with stochastically switching topologies. Our work reveals that the event/self-triggered algorithms are essentially discrete and thus more
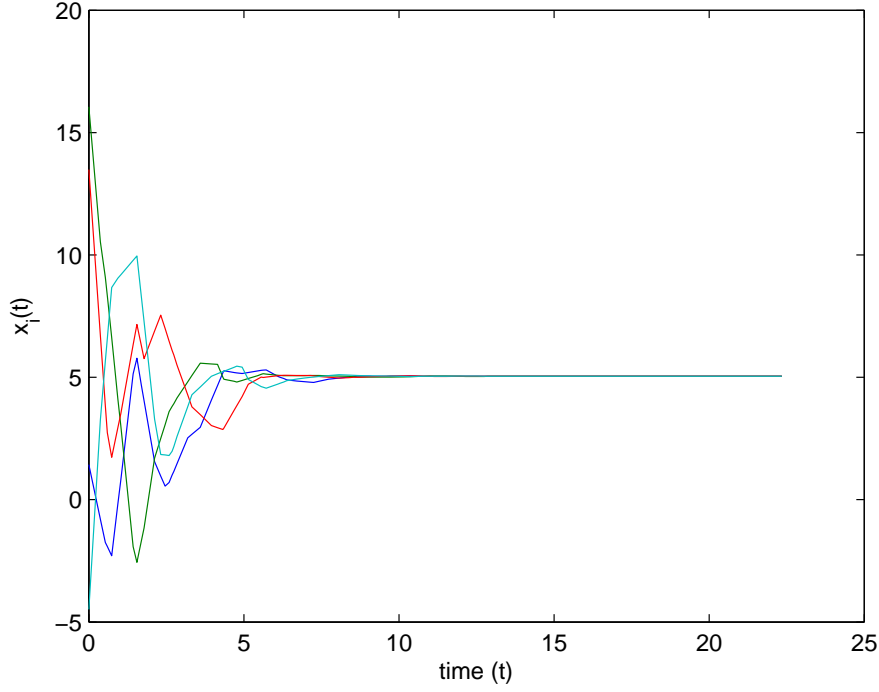
15

Fig. 1. Self-triggered consensus in networks of four agents with stochastically switching topologies.

suitable to a discrete analysis framework. And it is predictable that more results can be obtained for self-triggered algorithms under our new analysis framework in the future.

APPENDIX: DISCRETE-TIME CONSENSUS ANALYSIS

The appendix is devoted to the discussion of discrete-time consensus algorithms in networks of multi-agents with time delays based on the product theory of stochastic matrices and provide some results that is used in the main text.

A. Preliminaries

This subsection will provide some preliminaries on stochastic matrices that will be used in the proof of Theorem 6 in the next subsection.

A nonnegative matrix A is called a *stochastic indecomposable and aperiodic (SIA)* if it is a stochastic matrix and there exists a column vector v such that $\lim_{k \to \infty} A^k = \mathbf{e}_n v^\top$, where $\mathbf{e}_n$ is an n-dimensional column vector with all entries 1. A sequence of stochastic matrices $\{A_k\}$ is

16

ergodic if there exists a column vector v such that $\lim_{k \to \infty} \prod_{k=1}^{\infty} A_k = \mathbf{e}_n v^\top$. A is called δ-*SIA* if its δ-matrix is SIA. A stochastic matrix A is called *scrambling* if for any pair $(i,\ j)$, there exists an index k such that both $a_{ik} > 0$ and $a_{jk} > 0$. Similarly, A is called δ-*scrambling* if its δ-matrix is scrambling. For a matrix A, $\mathrm{diag}(A)$ refers to the diagonal matrix formed by the diagonal elements of A, i.e., $[\mathrm{diag}(A)]_{ii} = [A]_{ii}$ for all i and $[\mathrm{diag}(A)]_{ij} = 0$ for $i \neq j$. And we define $\overline{\mathrm{diag}}(A) = A - \mathrm{diag}(A)$.

We use $\mathbb{P}\{\cdot|\mathcal{F}\}$ to denote the conditional probability with respect to $\mathcal{F}$ on a probability space $\{\Omega, \mathbb{P}, \mathcal{F}\}$, which can also be expressed into the conditional expectation of an indicator function, i.e., $\mathbb{P}\{\cdot|\mathcal{F}\} = \mathbb{E}\{\mathbf{1}_{\{\cdot\}}|\mathcal{F}\}$, where $\mathbf{1}_{\{\cdot\}}$ is an indicator function which is defined as

$$\mathbf{1}_S(x) = \begin{cases} 1, & x \in S, \\ 0, & x \notin S. \end{cases}$$

Thus, $\mathbb{P}\{\cdot|\mathcal{F}\}$ is also a random variable that is measurable with respect to $\mathcal{F}$.

In the following, we will introduce some lemmas that will be used in the proof of the main results.

Let $A = [a_{ij}]$ be an $n \times n$ stochastic matrix. Define

$$\Delta(A) = \max_j \max_{i_1, i_2} |a_{i_1 j} - a_{i_2 j}|.$$

It can be seen that $\Delta(A)$ measures how different the rows of A are. $\Delta(A) = 0$ if and only if the rows of A are identical. Define

$$\lambda(A) = 1 - \min_{i_1, i_2} \sum_j \min\{a_{i_1 j}, a_{i_2 j}\}.$$

Remark 3: It can be seen that if A is scrambling, then $\lambda < 1$, if A is δ-scrambling for some $\delta > 0$, then $\lambda(A) \leq 1 - \delta$.

Lemma 3: [24] For any stochastic matrices $A_1, A_2, \cdots, A_k$, $k > 0$,

$$\Delta(A_1 A_2 \cdots A_k) \leq \prod_{i=1}^{k} \lambda(A_i).$$

Remark 4: From Lemma 3, it can be seen that for a sequence of stochastic matrices $\{A_k\}$, if there exists a sequence $\{k_i\}$ such that $k_i < k_{i+1}$, $\lim_{i \to \infty} k_i = \infty$, and $\prod_{i=1}^{\infty} \lambda(\prod_{m=k_i+1}^{k_{i+1}} A_m) = 0$, then $\lim_{k \to \infty} \Delta(\prod_{m=1}^{k} A_m) = 0$, i.e., $\{A_k\}$ is ergodic. Particularly, from the property of scrabling-ness, if there exists $\delta > 0$ such that there are infinite matrices in the sequence $\{\prod_{m=k_i+1}^{k_{i+1}} A_m\}$ that are δ-scrambling, then $\{A_k\}$ is ergodic.

17

Lemma 4: [24], [26] Let A_1, A_2, $\cdots$, A_k (repetitions permitted) be $r \times r$ SIA matrices with the property that for any $1 < k_1 < k_2 \leq k$, $\prod_{i=k_1}^{k_2} A_i$ is SIA. If $k > \mathbf{n}(r)$, then $\prod_{i=1}^{k} A_i$ is a scrambling matrix. Here $\mathbf{n}(r)$ is the number of different types of all $r \times r$ stochastic matrices.

Lemma 5: [26] Let A_1, $\cdots$, A_m be $n \times n$ nonnegative matrices, let

$$
D = \begin{bmatrix}
A_1 & A_2 & \cdots & A_m \\
0 & 0 & \cdots & 0 \\
\cdots & \cdots & \ddots & \cdots \\
0 & 0 & \cdots & 0
\end{bmatrix}_{mn \times mn},
$$

let

$$
M_0 = \begin{bmatrix}
I & 0 & \cdots & 0 & 0 \\
I & 0 & \cdots & 0 & 0 \\
0 & I & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & 0 & 0 \\
0 & 0 & \cdots & I & 0
\end{bmatrix}_{mn \times mn},
$$

and let $M_k = D + M_0^k$ for any $k \in \{1, 2, \cdots, m-1\}$. Then if $\mathcal{G}(\sum_{i=1}^{m} A_i)$ contains a spanning tree, then $\mathcal{G}(M_k)$ contains a spanning tree with the property that the root vertex of the spanning tree has a self-loop in $\mathcal{G}(M_k)$.

Remark 5: Actually, since $M_0^k = M_0^{m-1} = \begin{bmatrix} I & 0 & \cdots & 0 \\ I & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I & 0 & \cdots & 0 \end{bmatrix}$ for all $k \geq m-1$, thus Lemma 5 holds for all $k \in \mathbb{N}$. This is important for our further results based on this lemma.

Remark 6: If the condition is $\mathcal{G}(\sum_{i=1}^{m} A_i)$ has a δ-spanning tree for some $\delta > 0$, then the conclusion becomes that $\mathcal{G}(M_k)$ has a δ'-spanning tree whose root vertex has a self-loop, where $\delta' \geq \delta/m$.

Lemma 6: [25] Let A be a stochastic matrix. If $\mathcal{G}(A)$ contains a spanning tree with the property that the root vertex of the spanning tree has a self-loop in $\mathcal{G}(A)$, then A is SIA.

From Lemma 5 and 6, we can have the following Lemma.

Lemma 7: Let $A_1^i, \cdots, A_m^i$, $i = 1, \cdots, k$ be $n \times n$ nonnegative matrices. Let

$$
D_i = \begin{bmatrix} A_1^i & A_2^i & \cdots & A_m^i \\ 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}_{mn \times mn} ,
$$

and M_0 being defined as in Lemma 5. Then if $\mathcal{G}(\sum_{i=1}^k \sum_{j=1}^m A_j^i)$ has a δ-spanning tree for some $\delta > 0$, then the product $\prod_{i=1}^k (D_i + M_0)$ has a δ'-spanning tree for some $0 < \delta' < \delta$. And the root of the spanning tree has a self-loop, thus $\prod_{i=1}^k (D_i + M_0)$ is δ'-SIA.

Remark 7: It is obvious from Lemma 5 that if there exists $\epsilon > 0$ such that $D_i' \geq \epsilon(M_0 + D_i)$, then $\prod_{i=1}^k D_i'$ is δ'-SIA for some $0 < \delta' < \delta$. This is the case that will appear in the proof of Theorem 6.

Proof: First, we have

$$
\prod_{i=1}^k (M_0 + D_i) \geq M_0^k + \sum_{i=1}^k M_0^{k-i} D_i M_0^{i-1} \geq M_0^k + \sum_{i=1}^k D_i M_0^{i-1}, \tag{19}
$$

where the second inequality is due to the fact that $M_0^j D_i \geq D_i$ for any $j \geq 0$. And it is not difficult to verify that the first block row sum of D_i is preserved in $D_i M_0^{i-1}$ for $i = 1, \cdots, k$. Thus the first block row sum of $\sum_{i=1}^k D_i M_0^{i-1}$ equals that of $\sum_{i=1}^k D_i$, i.e., $\sum_{i=1}^k \sum_{j=1}^m A_j^i$. This means $\mathcal{G}(\sum_{i=1}^k D_i M_0^{i-1})$ has a δ-spanning tree. From Lemma 5, $\mathcal{G}(M_0^k + \sum_{i=1}^k D_i M_0^{i-1})$ has a δ'-spanning tree for some $0 < \delta' < \delta$, and the root has a self-loop. This is also true for the product $\prod_{i=1}^k (M_0 + D_i)$ due to (19). Thus, $\prod_{i=1}^k (M_0 + D_i)$ is δ'-SIA from Lemma 6. The proof is completed. ■

From Lemma 7, we can easily obtain the following corollary.

Corollary 1: Let $A_1^i, \cdots, A_m^i$, $i = 1, 2, \cdots$, be uniformly bounded $n \times n$ nonnegative matrices. Let D_i and M_0 be defined as in Lemma 7. If there exists k such that $\mathcal{G}(\sum_{i=1}^k \sum_{j=1}^m A_j^i)$ has a δ-spanning tree for some $\delta > 0$, then for each $k' > k$, the product $\prod_{i=1}^{k'} (D_i + M_0)$ is $\delta'(k')$-SIA, where $0 < \delta'(k') < \delta$ depends on k'.

Proof: Since $A_j^i \geq 0$, if $\mathcal{G}(\sum_{i=1}^k \sum_{j=1}^m A_j^i)$ has a δ-spanning tree for some $\delta > 0$, then for each $k' > k$, $\mathcal{G}(\sum_{i=1}^{k'} \sum_{j=1}^m A_j^i)$ also has a δ-spanning tree. The conclusion is obvious from Lemma 7. ■

Remark 8: If $A_1^i, \cdots, A_m^i$, $i = 1, 2, \cdots$, is a random sequence, then it is obvious from Corollary 1 that the event $\mathcal{G}(\sum_{i=1}^{k} \sum_{j=1}^{m} A_j^i)$ has a δ-spanning tree is contained in the event $\prod_{i=1}^{k'}(D_i + M_0)$ is $\delta'(k')$-SIA.

Lemma 8 (second Borel–Cantelli lemma [27]): Let $\{\mathcal{F}_n, n \geq 0\}$ be a filtration with $\mathcal{F}_0 = \{\emptyset, \Omega\}$ and $\{X_n, n \geq 1\}$ a sequence of events with $X_n \in \mathcal{F}_n$. Then

$$\{X_n \text{ occurs infinitely often }\} = \Big\{ \sum_{n=1}^{+\infty} \mathbb{P}\{X_n|\mathcal{F}_{n-1}\} = +\infty \Big\},$$

with a probability 1, where "infinitely often" means that an infinite number of events from $\{X_n\}_{n=1}^{\infty}$ occur.

Lemma 9: Let $\{A_k, \mathcal{F}_k\}$ be an adapted random sequence. If for a sequence $\{X_k\}$ defined as $X_k = \{A_k \in S_k\}$ for some given set S_k there exists $\delta > 0$ such that the conditional probability $\mathbb{P}\{X_{m+1}|\mathcal{F}_m\} > \delta$, then for each $m, h > 0$,

$$\mathbb{P}\{X_{m+1}, \cdots, X_{m+h}|\mathcal{F}_m\} > \delta^h.$$

Proof: By the definition of conditional probability, we have

$$\begin{aligned}
&\mathbb{P}\{X_{m+1}, \cdots, X_{m+h}|\mathcal{F}_m\} \\
=\ & \mathbb{E}\{\mathbf{1}_{X_{m+1}} \cdots \mathbf{1}_{X_{m+h}}|\mathcal{F}_m\} \\
=\ & \mathbb{E}\{\mathbf{1}_{X_{m+1}} \cdots \mathbb{E}\{\mathbf{1}_{X_{m+h}}|\mathcal{F}_{m+h-1}\}|\mathcal{F}_m\} \\
\geq\ & \delta\, \mathbb{E}\{\mathbf{1}_{X_{m+1}} \cdots \mathbf{1}_{X_{m+h-1}}|\mathcal{F}_m\} \\
\geq\ & \delta^2\, \mathbb{E}\{\mathbf{1}_{X_{m+1}} \cdots \mathbf{1}_{X_{m+h-2}}|\mathcal{F}_m\} \\
\geq\ & \cdots \\
\geq\ & \delta^h.
\end{aligned}$$

$\blacksquare$

Lemma 10: (Lemma 5.4 in [28]) Let $\{\Omega, \mathcal{F}, \mathbb{P}\}$ be a probability space, and f be a random variable with $0 \leq f \leq 1$. If for a σ-algebra $\mathcal{F}' \subseteq \mathcal{F}$, a set $S \in \mathcal{F}'$ with $\mathbb{P}\{S\} > 0$, $\mathbb{E}\{f|\mathcal{F}'\} \geq \delta$ holds on S for some $\delta > 0$, then we have

$$\mathbb{P}\{f \geq \frac{\delta}{2}|\mathcal{F}'\} \geq \frac{\delta}{2},$$

on S and particularly,

$$\mathbb{E}\{f\mathbf{1}_{\{f\geq\frac{\delta}{2}\}}|\mathcal{F}'\} \geq \frac{\delta^2}{4}$$

on S.

B. Discrete-time delayed consensus analysis

In this section, we will discuss discrete-time consensus algorithms in networks with switching topologies and time delays. We first consider general stochastic processes described by an adapted sequence and establish a most general result. Then we use it to obtain two corollaries under two special cases that will appear in the main text.

Consider the following discrete time dynamical systems with switching topologies:

$$x_i(k+1) = \sum_{l=0}^{\tau}\sum_{j=1}^{n} a_{ij}^l(k)x_j(k-l), \quad i = 1,\cdots,n, \tag{20}$$

where k is the time index, $x(k) = [x_1(k),\cdots,x_n(k)]^\top \in \mathbb{R}^n$ is the state variable of the system at time k, $\tau \geq 0$ is the bound of the time delay, where $\tau = 0$ corresponds to the case of no time delay. In the following, we always assume that $a_{ii}^0(k) > 0$, $a_{ii}^l(k) = 0$ for $l \neq 0$, $a_{ij}^l(k) \geq 0$ for each i, j, l, k, and

$$\sum_{l=0}^{\tau}\sum_{j=1}^{n} a_{ij}^l(k) = 1$$

holds for each k and i.

Define $A^l(k) = [a_{ij}^l(k)]$, and $B(k) = [b_{ij}(k)] = \sum_{l=0}^{\tau} A^l(k)$.

We have the following Theorem.

Theorem 6: Let $\{A^0(k),\cdots,A^\tau(k),\mathcal{F}_k\}$ be an adapted process. If there exists $\delta > 0$, $h > 0$ such that $B(k) \geq \delta I$ and the conditional expectation $\mathbb{E}\{\sum_{k=m+1}^{m+h} B(k)|\mathcal{F}_m\}$ has a δ-spanning tree, then the system (20) will reach consensus almost surely.

Remark 9: When $\tau = 0$, the system has no time delays, and Theorem 6 reduces to Theorem 3.1 of [28]. From this point of view, this result can be seen as an extension of that in [28].

The proof of this theorem will be divided into several steps.

First, we will prove the following lemma, which shows that the conditional expectation of a spanning tree implies a positive conditional probability of existence of a spanning tree for a longer length of the summation.

Lemma 11: Let $\{A_k, \mathcal{F}_k\}$ be an adapted random sequence of $n \times n$ stochastic matrices, if there exists $\delta > 0$ such that for each m, $\mathbb{E}\{A_{m+1}|\mathcal{F}_m\}$ has a δ -spanning tree, then there exist $h > 0$, $0 < \delta' < \delta$ such that

$$\mathbb{P}\{\sum_{k=m+1}^{m+h} A_k \text{ has a } \delta'\text{-spanning tree } |\mathcal{F}_m\} > \delta'. \tag{21}$$

Proof: Denote $\mathrm{Sp}(n)$ the number of different types of spanning trees composed of n vertices. For some fixed m, we decompose the probability space Ω with respect to $\mathcal{F}_m$, i.e., $S_m^{(i_1)} \in \mathcal{F}_m$, $\Omega = \cup S_m^{(i_1)}$, $i_1 = 1, 2, \cdots, s_1$ where $s_1 \leq \mathrm{Sp}(n)$, and $S_m^{(i)} \cap S_m^{(j)} = \emptyset$ for $i \neq j$ such that on each $S_m^{(i)}$, $\mathbb{E}\{A_{m+1} \mid \mathcal{F}_m\}$ has a (fixed) specified spanning tree. Denote $\mathbb{E}\{A_{m+1} \mid \mathcal{F}_m\}_{S_m^{(i_1)}}$ the restriction of $\mathbb{E}\{A_m \mid \mathcal{F}_m\}$ on $S_m^{(i_1)}$, i.e.,

$$\mathbb{E}\{A_{m+1} \mid \mathcal{F}_m\}_{S_m^{(i_1)}} = \mathbb{E}\{A_{m+1} \mid \mathcal{F}_m\} \times \mathbf{1}_{S_m^{(i_1)}}.$$

Let $\mathrm{ST}_\delta(\mathbb{E}\{A_{m+1} \mid \mathcal{F}_m\}_{S_m^{(i_1)}})$ be a δ-spanning tree contained in $\mathbb{E}\{A_{m+1} \mid \mathcal{F}_m\}_{S_m^{(i_1)}}$, which is arbitrarily selected when more than one choices are available. For each $S_m^{(i_1)}$, we pick an edge $(j, i) \in \mathcal{E}(\mathrm{ST}_\delta(\mathbb{E}\{A_{m+1} \mid \mathcal{F}_m\}_{S_m^{(i_1)}}))$, and let $S_{m+1}^{(i_1)} = \{[A_{m+1}]_{ij} \geq \delta/2\} \cap S_m^{(i_1)}$. Then $S_{m+1}^{(i_1)} \in \mathcal{F}_{m+1}$, and from Lemma 10,

$$\mathbb{P}\{S_{m+1}^{(i_1)} \mid \mathcal{F}_m\} \geq \frac{\delta}{2}$$

holds on $S_m^{(i_1)}$.

Similarly, we decompose each $S_{m+1}^{(i_1)}$ with respect to $\mathcal{F}_{m+1}$, i.e., $S_{m+1}^{(i_1)} = \cup_{i_2} S_{m+1}^{(i_1 i_2)}$ with $S_{m+1}^{(i_1 i_2)} \in \mathcal{F}_{m+1}$ and $S_{m+1}^{(i_1 i_2)} \cap S_{m+1}^{(i_1 i_2')}$ for $i_2, i_2' = 1, 2, \cdots, s_2$, $i_2 \neq i_2'$, where $s_2 \leq \mathrm{Sp}(n)$ depends on the index i_1 such that $\mathbb{E}\{A_{m+2} \mid \mathcal{F}_{m+1}\}_{S_{m+1}^{(i_1 i_2)}}$ has a specified δ-spanning tree on each $S_{m+1}^{(i_1 i_2)}$.

For each $S_{m+1}^{(i_1 i_2)}$, we pick an edge $(j, i) \in \mathcal{E}(\mathrm{ST}_\delta(\mathbb{E}\{A_{m+2} \mid \mathcal{F}_{m+1}\}_{S_{m+1}^{(i_1 i_2)}}))$ and let $S_{m+2}^{(i_1 i_2)} = \{[A_{m+2}]_{ij} \geq \delta/2\} \cap S_{m+1}^{(i_1 i_2)}$, then $S_{m+2}^{(i_1 i_2)} \in \mathcal{F}_{m+2}$ and from Lemma 10,

$$\mathbb{P}\{S_{m+2}^{(i_1 i_2)} \mid \mathcal{F}_{m+1}\} \geq \frac{\delta}{2}$$

holds on $S_{m+1}^{(i_1 i_2)}$.

Continuing this process, we can get a sequence:

$$S_m^{(i_1)}, S_{m+1}^{(i_1)}, S_{m+1}^{(i_1 i_2)}, S_{m+2}^{(i_1 i_2)}, \cdots, S_{m+k-1}^{(i_1 i_2 \cdots i_k)}, S_{m+k}^{(i_1 i_2 \cdots i_k)}$$

such that for $l = 1, 2, \cdots, k$, $S_{m+l}^{(i_1 i_2 \cdots i_l)} \in \mathcal{F}_{m+l}$, $S_{m+l-1}^{(i_1 i_2 \cdots i_{l-1})} = \cup_{i_l} S_{m+l-1}^{(i_1 i_2 \cdots i_l)}$, $\mathbb{E}\{A_{m+l} \mid \mathcal{F}_{m+l-1}\}_{S_{m+l-1}^{(i_1 i_2 \cdots i_l)}}$ has a (fixed) spanning tree, and

$$S_{m+l}^{(i_1 i_2 \cdots i_l)} = \{[A_{m+l}]_{ij} > \delta/2\} \cap S_{m+l-1}^{i_1 i_2 \cdots i_l}, \tag{22}$$

22

where the edge $(j, i) \in \mathcal{E}(\mathrm{ST}_\delta(\mathbb{E}\{A_{m+l} \mid \mathcal{F}_{m+l-1}\}_{S_{m+l-1}^{(i_1 i_2 \cdots i_l)}}))$.

For any fixed sequence $i_1, i_2, \cdots, i_k$, $S_{m+k}^{(i_1 i_2 \cdots i_k)} \subseteq S_{m+k-1}^{(i_1 i_2 \cdots i_k)} \subseteq S_{m+k-1}^{(i_1 i_2 \cdots i_{k-1})} \subseteq \cdots \subseteq S_{m+1}^{(i_1)}$. We choose the edge (j, i) in (22) for each $S_{m+l}^{(i_1 i_2 \cdots i_l)}$ in such way that if $\mathrm{ST}_\delta(\mathbb{E}\{A_{m+1} \mid \mathcal{F}_{m+l-1}\}_{S_{m+l-1}^{(i_1 i_2 \cdots i_l)}})$ hasn't appeared in the sequence, $\mathrm{ST}_\delta(\mathbb{E}\{A_{m+1} \mid \mathcal{F}_m\}_{S_m^{(i_1)}}), \cdots, \mathrm{ST}_\delta(\mathbb{E}\{A_{m+1-1} \mid \mathcal{F}_{m+l-2}\}_{S_{m+l-2}^{(i_1 i_2 \cdots i_{l-1})}})$, then we choose $(j, i) \in \mathcal{E}(\mathrm{ST}_\delta(\mathbb{E}\{A_{m+1} \mid \mathcal{F}_{m+l-1}\}_{S_{m+l-1}^{(i_1 i_2 \cdots i_l)}}))$ arbitrarily. Otherwise, we choose $(j, i) \in \mathcal{E}(\mathrm{ST}_\delta(\mathbb{E}\{A_{m+1} \mid \mathcal{F}_{m+l-1}\}_{S_{m+l-1}^{(i_1 i_2 \cdots i_l)}}))$ which hasn't been chosen before when $\mathrm{ST}_\delta(\mathbb{E}\{A_{m+1} \mid \mathcal{F}_{m+l-1}\}_{S_{m+l-1}^{(i_1 i_2 \cdots i_l)}})$ appears in the sequence $\mathrm{ST}_\delta(\mathbb{E}\{A_{m+1} \mid \mathcal{F}_m\}_{S_m^{(i_1)}}), \cdots, \mathrm{ST}_\delta(\mathbb{E}\{A_{m+1-1} \mid \mathcal{F}_{m+l-2}\}_{S_{m+l-2}^{(i_1 i_2 \cdots i_{l-1})}})$ if there still exists such an edge.

Since there at most $\mathrm{Sp}(n)$ different types of spanning trees, and each spanning tree has $n - 1$ edges, it is obvious when $k = (n - 1)\mathrm{Sp}(n)$, the sets

$$S_{m+k}^{(i_1 i_2 \cdots i_k)} \subseteq \{\sum_{l=1}^{k} A_{m+l} \text{ has } \delta/2\text{-spanning tree}\}.$$

This is because for each fixed sequence $i_1, \cdots, i_k$, at least one type of spanning trees has appeared at least $n - 1$ times, thus by the edge chosen strategy, each of its edges has been chosen at least once.

So we have

$$\mathbb{P}\{\sum_{l=1}^{k} A_{m+l} \text{ has } \delta/2\text{-spanning tree} \mid \mathcal{F}_m\} \tag{23}$$

$$\geq \mathbb{P}\{\cup_{i_1,i_2,\cdots,i_k} S_{m+k}^{(i_1 i_2 \cdots i_k)} \mid \mathcal{F}_m\}$$

$$= \mathbb{E}\{\sum_{i_1,i_2,\cdots,i_k} \mathbf{1}_{S_{m+k}^{(i_1 i_2 \cdots i_k)}} \mid \mathcal{F}_m\}$$

$$= \mathbb{E}\{\mathbb{E}\{\sum_{i_1,i_2,\cdots,i_k} \mathbf{1}_{S_{m+k}^{(i_1 i_2 \cdots i_k)}} \mid \mathcal{F}_{m+k-1}\} \mid \mathcal{F}_m\} \tag{24}$$

$$\geq \frac{\delta}{2} \mathbb{E}\{\sum_{i_1,i_2,\cdots,i_k} \mathbf{1}_{S_{m+k-1}^{(i_1 i_2 \cdots i_k)}} \mid \mathcal{F}_m\} \tag{25}$$

$$= \frac{\delta}{2} \mathbb{E}\{\sum_{i_1,i_2,\cdots,i_{k-1}} \mathbf{1}_{S_{m+k-1}^{(i_1 i_2 \cdots i_{k-1})}} \mid \mathcal{F}_m\} \tag{26}$$

$$\geq \cdots$$

$$\geq (\frac{\delta}{2})^{k-1} \mathbb{E}\{\sum_{i_1} \mathbf{1}_{S_{m+1}^{(i_1)}} \mid \mathcal{F}_m\}$$

$$\geq (\frac{\delta}{2})^{k} \mathbb{E}\{\sum_{i_1} \mathbf{1}_{S_{m}^{(i_1)}} \mid \mathcal{F}_m\}$$

$$= (\frac{\delta}{2})^{k},$$

The inequality from (24) to (25) is due to the fact

$$\mathbb{P}\{S_{m+k}^{(i_1 i_2 \cdots i_k)} \mid \mathcal{F}_{m+k-1}\} \geq \delta/2$$

on $S_{m+k-1}^{(i_1 i_2 \cdots i_k)} \in \mathcal{F}_{m+k-1}$, which implies

$$\mathbb{E}\{\mathbf{1}_{S_{m+k}^{(i_1 i_2 \cdots i_k)}} \mid \mathcal{F}_{m+k-1}\} \geq \frac{\delta}{2} \mathbf{1}_{S_{m+k-1}^{(i_1 i_2 \cdots i_k)}}.$$

The equality from (25) to (26) is due to the fact that $S_{m+k-1}^{(i_1 i_2 \cdots i_{k-1})} = \cup_{i_k} S_{m+k-1}^{(i_1 i_2 \cdots i_k)}$, and $S_{m+k-1}^{(i_1 i_2 \cdots i_{k-1} j)} \cap S_{m+k-1}^{(i_1 i_2 \cdots i_{k-1} j')} = \emptyset$ for $j \neq j'$, which implies

$$\mathbf{1}_{S_{m+k-1}^{(i_1 i_2 \cdots i_{k-1})}} = \sum_{i_k} \mathbf{1}_{S_{m+k-1}^{(i_1 i_2 \cdots i_{k-1} i_k)}}$$

for each fixed sequence $i_1, i_2, \cdots, i_{k-1}$.

Let $h = (n-1)\mathrm{Sp}(n)$, and $\delta' = (\delta/2)^h$, then (21) holds. The proof is completed. ∎

Now we come to the proof of Theorem 6.

Proof of Theorem 6

24

Denote $y_k = [x(k), x(k-1), \cdots, x(k-\tau)]^\top$, then we have:

$$y_{k+1} = C_k y_k, \tag{27}$$

where

$$C_k = \begin{bmatrix} A^0(k) & A^1(k) & \cdots & A^{\tau-1}(k) & A^\tau(k) \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix}.$$

Let $B'_{k'} = \sum_{k=(k'-1)h+1}^{k'h} B_k$, and $\mathcal{F}'_{k'} = \mathcal{F}_{k'h}$. Then $\{B'_k, \mathcal{F}'_k\}$ also forms an adapted sequence, and $\mathbb{E}\{B'_{m+1} \mid \mathcal{F}'_m\}$ has a δ-spanning tree. From Lemma 11, there exists $\delta' > 0$ such that

$$\mathbb{P}\left\{ \sum_{k=m+1}^{m+(n-1)\operatorname{Sp}(n)} B'_k \text{ has a } \delta'\text{-spanning tree} \mid \mathcal{F}'_m \right\} > \delta'.$$

Since

$$C_k = \begin{bmatrix} \operatorname{diag}(A^0(k)) & 0 & \cdots & 0 & 0 \\ I & 0 & \cdots & 0 & 0 \\ 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \end{bmatrix} + \begin{bmatrix} \overline{\operatorname{diag}}(A^0(k)) & A^1(k) & \cdots & A^{\tau-1}(k) & A^\tau(k) \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

$$\geq \quad \delta M_0 + D_k \geq \delta(M_0 + D_k),$$

where D_k refers to the second matrix on the righthand side of the first line. From Lemma 7, Corollary 1, and Remark 7, Remark 8, there exist $\delta''(p) \in (0, \delta')$, $p \geq 1$ such that

$$\mathbb{P}\left\{ \prod_{k=m+1}^{m+p(n-1)\operatorname{Sp}(n)} C_k \text{ is } \delta''(p)\text{-SIA}, p = 1, 2, \cdots \mid \mathcal{F}'_m \right\} > \delta'.$$

Let $C'_{k'} = \prod_{k=(k'-1)(n-1)\operatorname{Sp}(n)+1}^{k'(n-1)\operatorname{Sp}(n)} C_k$, $\mathcal{F}''_{k'} = \mathcal{F}'_{k'(n-1)\operatorname{Sp}(n)}$, then $\{C'_k, \mathcal{F}''_k\}$ forms an adapted sequence and

$$\mathbb{P}\left\{ \prod_{k=1}^{p} C'_{m+k} \text{ is } \delta''(p)\text{-SIA }, p = 1, 2, \cdots \mid \mathcal{F}''_m \right\} > \delta'.$$

Let $k_n = \mathbf{n}(n) + 1$, then from Lemma 4 and Lemma 9,

$$\mathbb{P}\Big\{\prod_{i=1}^{k_n} C'_{m+i} \text{ is } \delta''(1)^{k_n}\text{-scrambling}\big|\ \mathcal{F}''_m\Big\}$$

$$\geq\ \mathbb{P}\Big\{\prod_{k=1}^{p} C'_{m'+k} \text{ is } \delta''(p)\text{-SIA},\ m' = m, \cdots, m + k_n - 1, p = 1, 2, \cdots .|\ \mathcal{F}''_m\Big\}$$

$$>\ \delta'^{k_n}.$$

Let $C'''_m = \prod_{k=(m-1)k_n+1}^{mk_n} C'_k$, and $\mathcal{F}'''_m = \mathcal{F}''_{mk_n}$. Then $\{C'''_k, \mathcal{F}'''_k\}$ forms an adapted sequence, and

$$\mathbb{P}\{C'''_{m+1} \text{ is } \delta''(1)^{k_n}\text{-scrambling} \mid \mathcal{F}'''_m\} > \delta'^{k_n}.$$

From the Second Borel-Cantelli Lemma (Lemma 8), this implies

$$\mathbb{P}\{C'''_k \text{ is } \delta''(1)^{k_n}\text{-scrambling infinitely often}\} = 1.$$

From Lemma 3 and Remark 4, we have

$$\mathbb{P}\Big\{\lim_{k \to +\infty} \Delta\Big(\prod_{m=1}^{k} C_m\Big) = 0\Big\} = 1.$$

This implies the system (27) reaches consensus almost surely. On the other hand it is not difficult to show that consensus of (27) can imply consensus of (20). And the proof is completed.

From Theorem 6, we can have the following two corollaries. The first one is for deterministic case.

Corollary 2: Let $\{A^0(k), \cdots, A^\tau(k)\}$ be a deterministic sequence with $A^0(k) \geq \delta I$ for some $\delta > 0$, and $[A^i(k)]_{jj} = 0$ for $i = 1, 2, \cdots, \tau$, $j = 1, \cdots, n$. If there exists $h > 0$ such that $\sum_{k=m+1}^{m+h} B(k)$ has a δ-spanning tree, then the system (20) will reach consensus.

The second one is for a random sequence that will become independent after a fixed length of time.

Corollary 3: Let $\{A^0(k), \cdots, A^\tau(k)\}$ be a random sequence with $A^0(k) \geq \delta I$ for some $\delta > 0$, and $[A^i(k)]_{jj} = 0$ for $i = 1, 2, \cdots, \tau$, $j = 1, \cdots, n$. And there exists $N > 0$ such that $A^0(k), \cdots, A^\tau(k)$ is independent of $A^0(k'), \cdots, A^\tau(k')$ whenever $k' \geq k + N$. If there exists $h > 0$ such that $\sum_{k=m+1}^{m+h} \mathbb{E}\, B(k)$ has a δ-spanning tree, then the system (20) will reach consensus almost surely.

Proof: Let $\mathcal{F}_k = \sigma\{A^0(m), \cdots, A^\tau(m), m \leq k\}$ be the σ-algebra formed by $A^0(m), \cdots, A^\tau(m)$, $m \leq k$. Then $\{A^0(k), \cdots, A^\tau(k), \mathcal{F}_k\}$ forms an adapted process. From the assumption on

$\{A^0(k), \cdots, A^\tau(k)\}$, $A^0(k'), \cdots, A^\tau(k')$ is independent of $\mathcal{F}_k$ whenever $k' \geq k + N$. Since $B(k) \geq 0$, we have

$$\mathbb{E}\{\sum_{m=k+1}^{k+N+h} B(m)|\mathcal{F}_k\} \geq \mathbb{E}\{\sum_{m=k+N+1}^{k+N+h} B(m)|\mathcal{F}_k\} = \mathbb{E}\{\sum_{m=k+N+1}^{k+N+h} B(m)\} = \sum_{m=k+N+1}^{k+N+h} \mathbb{E}\, B(m).$$

Thus, $\mathbb{E}\{\sum_{m=k+1}^{k+N+h} B(m)|\mathcal{F}_k\}$ has a δ-spanning tree and the conclusion follows from Theorem 6. ∎

REFERENCES

[1] R. Olfati-Saber, and R.M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520-1533, Sep. 2004.

[2] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness", *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693-703, Aug. 2007.

[3] W. Ren and E.M. Atkins, "Distributed multi-vehicle coordinated control via local information exchange", *International Journal of Robust and Nonlinear Control*, vol. 17, no. 1011, pp. 1002-1033, 2007.

[4] M. Cao, B.D.O. Anderson, A.S. Morse, and C. Yu, "Control of acyclic formations of mobile autonomous agents," in *Proceedings of the 47th IEEE Conference on Decision and Control*, 2008, 1187-1192.

[5] L. Consolini, F. Morbidi, D. Prattichizzo, and M. Tosques, "Leader-follower formation control of nonholonomic mobile robots with input constraints," *Automatica*, vol. 44, no. 5, pp. 1343-1349, 2008.

[6] R. Olfati-Saber and J.S. Shamma, "Consensus filters for sensor networks and distributed sensor fusion," in *Proceedings of the 44th IEEE Conference on Decision and Control*, 2005, pp. 6698-6703.

[7] A. Speranzon, C. Fischione, and K.H. Johansson, "Distributed and collaborative estimation over wireless sensor networks," in *Proceedings of the 45th IEEE Conference and Control*, 2006, pp. 1025-1030.

[8] P. Tabuada, "Event-triggered real-time scheduling of stablizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680-1685, Sep. 2007

[9] W.P.M.H. Heemels, J.H. Sandee, and P.P.J. Van Den Bosch, "Analysis of event-driven controllers for linear systems," *International Journal of Control*, vol. 81, no. 4, pp. 571-590, 2007.

[10] X. Wang and M.D. Lemmon, "Event design in event-triggered feedback control systems," in *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 2105-2110, 2008.

[11] X. Wang and M.D. Lemmon, "Event-triggering distributed networked control systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 586-601, Mar. 2011.

[12] M.J. Manuel and P. Tabuada, "Decentralized event-triggered control over wireless sensor/actuator networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 10, pp. 2456-2461, Oct. 2011.

[13] D.V. Dimarogonas, E. Frazzoli, and K.H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1291-1297, May 2012.

[14] Z. Liu, Z. Chen, and Z. Yuan, "Event-triggered average consensus of multi-agent systems with weighted and directed topology," *Journal of Systems Science and Complexity*, vol. 25, no. 5, pp. 845-855, 2012.

[15] G.S. Seyboth, D.V. Dimarogonas, and K.H. Johansson, "Event-based broadcasting for multi-agent average consensus," *Automatica*, vol. 49, pp. 245-252, 2013.

[16] Y. Fan, G. Feng, Y. Wang, and C. Song, "Distributed event-triggered control of multi-agent systems with combinational measurements," *Automatica*, vol. 49, pp. 671-675, 2013.

[17] W. L. Lu and T. Chen. "Synchronization analysis of linearly coupled networks of discrete time systems", *Physica D*, vol. 198, pp. 148-168, 2004.

[18] W. L. Lu and T. Chen. "Global Synchronization of Discrete-Time Dynamical Network With a Directed Graph", *IEEE Transactions on Circuits and Systems-II: Express Briefs*, vol. 54, no. 2, pp. 136-140, 2007.

[19] A. Anta, and P. Tabuada, "Self-triggered stabilization of homogeneous control systems," in *Proceedings of the American Control Conference*, 2008, pp. 4129-4134.

[20] M. Mazo, and P. Tabuada, "On event-triggered and self-triggered control over sensor/actuator networks," in *Proceedings of the 47th IEEE Conference on Decision and Control*, 2008, 435-440.

[21] X. Wang and M.D. Lemmon, "Self-triggered feedback control systems with finite-gain $\mathcal{L}_2$ stability," *IEEE Transactions on Automatic Control*, vol. 45, no. 3, pp. 452-467, Mar. 2009.

[22] M.J. Manuel, A. Anta, and P. Tabuada, "An ISS self-triggered iplementation of linear controllers," *Automatica*, vol. 46, no. 8, pp. 1310-1314, 2014.

[23] A. Anta and P. Tabuada, "To sample or not to sample: self-triggered control for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2030-2042, Sep. 2010.

[24] J. Wolfowitz, "Products of indecomposable, aperiodic, stochastic matrices", *Proceedings of the American Mathematical Society*, vol. 14 pp. 733-737, 1963.

[25] F. Xiao, and L. Wang, "State consensus for multi-agent systems with switching topologies and time-varying delays", *International Journal of Control*, vol. 79, pp. 1277-1284, 2006.

[26] F. Xiao, and L. Wang, "Asynchronous consensus in continuous-time multi-agent systems with switching topology and time-varying delays", *IEEE Transactions on Automatic Control*, vol. 53, no. 8, pp. 1804-1816, Sep 2008.

[27] R. Durrett, *Probability: Theory and Examples*, 3rd ed., Duxbury Press, Belmont, CA, 2005.

[28] B. Liu, W.L. Lu, and T.P. Chen, "Consensus in networks of multiagents with switching topologies modeled as adapted stochastic processes", *SIAM Journal on Control and Optimization*, vol. 49, no. 1, pp. 227-253, 2011.

[29] D.V. Dimarogonas, and K.J. Kyriakopoulos, "A connection between formation infeasibility and velocity alignment kinematic multi-agent systems," *Automatica*, vol. 44, no. 10, pp. 2648-2654, 2008.

[30] J.A. Fax, and R.M. Murray, "Graph Laplacians and stablization of vehicle formations", in *Proceedings of the 15th IFAC World Congress*, 2002, [CD ROM].